Evolved multi-agent systems and thorough evaluation are necessary for scalable logistics

(Position Paper)

Rinde R.S. van Lon
iMinds-DistriNet, dept. of Computer Science
KU Leuven, 3001 Leuven, Belgium
Rinde.vanLon@cs.kuleuven.be

Tom Holvoet
iMinds-DistriNet, dept. of Computer Science
KU Leuven, 3001 Leuven, Belgium
Tom.Holvoet@cs.kuleuven.be

Abstract—We consider logistics problems that are both dynamic and potentially large scale. A common way to create a scalable solution for logistics problems is to use multi-agent systems. In this paper we take two positions of a different nature: (1) evolutionary designed multi-agent systems are a promising approach to create scalable and performant solutions for logistics problems, (2) the multi-agent systems field does not prioritize evaluation enough, which hinders thorough scientific comparisons and prevents adoption in industry. We present arguments and refute common counterarguments for our position. Further, we discuss our present and upcoming efforts to realize our position.

I. INTRODUCTION

In logistics and transportation there are many dynamic factors. Most logistics companies handle the dynamic properties of their business by imposing constraints on their customers. For example, most transportation companies require you to place your order before a certain deadline, this allows the company to treat all orders as a batch that can be planned in one step. There are specific areas such as taxi services and courier services that are specialized in operating under very dynamic conditions. In this paper we consider the general dynamic pickup-and-delivery problem (PDP) \[1\] that includes taxi and courier services. There are generally two approaches for solving this class of problems: (1) formulate the problem as a sequence of static problems and solve it using state-of-the-art optimization algorithms (exact methods, heuristics, meta-heuristics), or (2) embracing dynamism either by approximation methods or by modeling the system as a decentralized multi-agent system (MAS).

Based on our experience with industry we infer that there are few multi-agent systems actually deployed at logistics companies. The two main reasons for this situation are: (1) multi-agent systems are often not optimized for a specific problem, (2) multi-agent systems can give no guarantees of their performance due to their decentralized nature. A cause for this state of affairs is that multi-agent systems are notoriously hard to develop such that they perform predictably at all times. In fact, the prevalent way to investigate the quality of a MAS is to evaluate it empirically on data. Engineering MAS is hard due to the difference between the local and the global level in a MAS. The local behaviors of the agents influence the emergent global properties of the system in a potentially non-linear way.

As of yet there is no established method for engineering non-trivial multi-agent systems that controls or gives guarantees about its emergent properties. As an effect, most agents use traditional solution strategies that seem to be reasonable to the developers. But whether or not these solutions are optimal or even efficient is often left without investigation.

Our position is that multi-agent systems can be greatly enhanced when using evolutionary computing \[2\] to develop them offline, and then deploy them online. Using evolution we believe the two earlier mentioned problems with engineering multi-agent systems can be overcome: (1) with evolution a MAS is automatically optimized for a specific problem and (2) evolution can be used with an objective function that describes an emergent property of the desired system. For example, a MAS can be created that is robust with respect to disruption. All that is required is an exact definition of robustness in the application domain. This approach does not solve the emergence problem entirely but it provides a method to limit any undesired behaviors.

There exist many (multi-agent system) solutions for dynamic logistics problems. Unfortunately, there does not, to our knowledge, exist a standard set of benchmarks and evaluation criteria on which a new solution can be tested. In this paper we take the position that establishing such a standard is instrumental for the further development of research in dynamic logistics problems.

The paper is structured as follows. In section II we explain our two positions and we refute common counterarguments against our positions in section III. We give a brief overview of related work in the fields of dynamic logistics, multi-agent systems and evolutionary algorithms in section IV. We discuss the work that is already and still needs to be done in order to realize our positions in section V. The conclusion is presented in section VI.

II. POSITION

In this section we present our two positions.

A. Evolution is better at designing MAS

In this section we argue why evolutionary computing combined with multi-agent systems is better compared to tradi-
tional hand constructed centralized algorithms for logistics.

1) **Inspiration from nature:** The inspiration for using evolution to develop decentralized systems comes from nature, more specifically from social insects such as ants, termites and bees. These social insects are remarkable in that they are self-organized, there is no central authority that controls all individuals. Yet, these organisms can perform complex tasks and can survive in a wide variety of environments. These organisms, themselves being the product of millions of years of evolution show that evolution can produce effective decentralized agent organizations when using an objective such as ‘survival’.

For logistics companies the objective is usually directly related to making profits. Profits allow a company to make new investments, to hire more people and to expand into new markets. These actions improve the success (or survival probabilities) of a company. In other words, it can be said that natural evolution for survival and artificial evolution for profit are remarkably similar. This similarity suggests that artificial evolution might also be capable of evolving decentralized systems for logistics companies.

2) **Decentralization for scalability and handling dynamism:** Due to the absence of a centralized entity, multi-agent systems are theoretically very good at scaling with the size of a problem. Problems can be solved locally (potentially by communicating with other strictly local agents). This mechanism can allow MAS to embrace dynamism in the environment. This is opposed to traditional centralized optimization algorithms which typically scale badly.

3) **Optimization by design:** A requirement for using evolution is the existence of data (real or simulated) that represents the problem to be solved. Most companies already possess this information, usually in the form of historical data. With this approach an evolved multi-agent system will be optimized by design to the available data of the application domain. This is in contrast to hand constructed multi-agent systems, where developers need to rely on assumptions about the application domain. This opens the way for MASs to match the performance of more traditional optimization techniques. This can in turn lead to a more general acceptance of multi-agent systems in logistics.

4) ** Emergence as a feature:** We are currently not aware of a set of guidelines or a methodology that helps controlling emergence in multi-agent systems. A first step to a more general methodology has been taken in [3]. By following this process one still has to make choices at the individual agent level that can influence the global level in unforeseen ways.

Adopting artificial evolution as the ‘designer’ of multi-agent systems evades this problem by making it part of the trial-and-error process that is evolution. By stating the desired emergent global behavior in an objective function evolution can search for a solution that matches the objective. Afterwards, using the same objective function, it can be validated whether the resulting system does indeed show the desired emergent behavior.

5) **Flexible and reusable:** A great advantage of using evolution for developing a MAS is that once everything is set up, it can be repeated by just running the algorithm again on the data. This allows companies to update their multi-agent systems when new knowledge or data is available for the problem. For example, changing fuel or personnel costs can be taken into account by slightly modifying the objective function. This means that developing a MAS is no longer a tedious task that requires numerous and expensive experts but that it can be repeated by a single push of a button at the cost of mere computing resources.

### Better evaluation is essential

In section IV we describe the related work on multi-agent systems in logistics. It is remarkable that of these papers there are only a few that do a thorough evaluation. Even the approaches that are thoroughly evaluated are often hard to compare with other works due to the lack of a standard benchmark. The lack of thorough evaluation reduces the scientific value of the field. We believe that a strong scientific basis is required in order for MASs to be accepted in industry. Therefore, our position is: a thorough evaluation should become imperative in multi-agent systems papers for logistics as this will improve the scientific quality of the field and ultimately boost its impact in industry.

To support our position we identified several issues that need to be addressed in order to facilitate evaluations in multi-agent systems for logistics:

1) **Standard simulator:** Different assumptions that are made in simulators can have a significant effect on simulation experiments, but these effects are notoriously hard to quantify. Therefore, a standard multi-agent simulator for a specific application domain, in this case dynamic logistics problems, would be beneficial for making valid comparisons. A standard simulator that is used by different research groups would significantly lower the barrier for other researchers to target the same problem. Further, using the same simulator ensures a leveled playing field.

2) **Benchmark datasets:** A standard set of realistic benchmark datasets should be made available on which every new idea should be tested in the domain of dynamic pickup-and-delivery problems. Preferably, the data should be based on real world data and should be accompanied by a reference implementation using the standard simulator. Having a standard set of benchmarks with implementations lowers the barrier for doing experiments even further. For example, a competition can be organized around the set of benchmarks, which can be helpful for improving the state-of-the-art.

3) **Evaluation metrics:** Each dataset should be accompanied by at least one standard evaluation metric. One of the problems in comparing different solutions is that they use different metrics. If a generally accepted set of metrics is used, evaluation is made much simpler, and progress is made easier.

4) **Open access:** To facilitate reproducibility it is instrumental to have open access to the standard simulator, datasets and metrics. Ideally, these components should not only be open
access but also open source since that allows reusability by others. Also, open sourcing all code greatly improves the trust and quality of software. We are not alone in this position, e.g. see [4].

III. COUNTERARGUMENTS

There exist several counterarguments for the two positions that we present in this paper. In this section we rebut these counterarguments by disputing their original claims and explaining how they can be overcome.

A. Counterarguments for evolution and MAS

There are several arguments that oppose the view that evolutionary computing can solve the problems faced by hand crafting multi-agent systems. In this section we refute three counterarguments that we deem to be the most significant.

1) Centralization versus decentralization: To reach a global optimum it is imperative that some part of the system holds a global view on the problem. Although this is true for a strictly static problem, this is not necessarily the case for dynamic problems. In fact, optimizing a solution using all currently available information in a dynamic problem can lead to sub-optimal performance due to local (in time) over-optimization that is harmful in light of frequent changes in the problem. In [5] it was shown that a solution that prioritizes optimization of early constraints performs better compared to solutions trying to optimize all known constraints at a time. Furthermore, a decentralized approach scales much better than a strictly centralized approach.

2) No hard guarantees: Self-organizing emergent multi-agent systems are notorious for their unpredictable nature and lack of guarantees of behavior. Unfortunately, using evolution these shortcomings can not (yet) be completely removed. However, we argue that using evolution provides us with much more empirical evidence that the resulting system does not show any undesirable behavior, although there is no strong guarantee. We present three ways of overcoming this problem:

a) Pure evolution: The easiest way to deal with this problem is to add a penalty in the objective function that penalizes undesirable (emergent) behavior. Also, it is important to add specific scenarios in the train set that is used during evolution that have a high probability of undesirable (emergent) behavior. An example in PDP can be to add a scenario with extreme vehicle failures to ensure the graceful degradation of the system. By evolving and testing on a large set of historical data this approach should give a reasonable indication of the quality of the solution.

b) Limiting evolution: In some cases, where the previous solution is not enough, a company might want to ensure that some things can never happen. In this case, there is the option of partially hand constructing an agent and using evolution to fill in the gaps. An example of this approach in a transportation case can be that an agent is disallowed to drive in circles. Inside the agent it can be easily checked whether this is attempted and take appropriate action.

c) Evolution as inspiration: In certain sensitive domains (e.g. nuclear power plants, space or security), or when people do not feel comfortable to place so much trust on this technology, the outcome of the evolutionary process can be used as input for a hand constructed multi-agent system. The evolved solutions can be reverse engineered such that they can be understood by humans. The analysis can be done on a behavioral level, by studying the MAS in a simulator on several scenarios. In case genetic programming is used the resulting programs can even be analyzed directly. In this way, evolved solutions can serve as an inspiration to improve hand crafted MASs. Even though evolution has a more limited role in this approach compared to the two previous approaches, it is still a significant factor in the developmental process.

3) Evolution is slow: Artificial evolution, like natural evolution, is not known for its speed of developing ‘systems’. However, a strong point of evolutionary algorithms is their ease of parallelization meaning that evolutionary algorithms directly benefit from the trend of the increasing number of cores in hardware. Since it is expected that this trend will continue and in the same time make computing cheaper the result will be that evolving multi-agent systems will in most cases be significantly cheaper compared to manually constructing them.

4) Using evolution does not solve all problems: This is an often heard objection that holds for evolutionary algorithms in general. It can be said that adopting evolution for a problem does not immediately solve the problem but rather transforms the original problem into an evolutionary design problem. A non-trivial evolutionary design decision is for example the design of a good genetic representation. Besides, many ‘magic parameters’ need to be determined, for example, population size, number of generations, crossover rate, mutation rate, etc. However, there exists an active research community around these techniques that can greatly help in this regard. Furthermore, evolution has shown to be able to outperform humans in many areas [6], which makes it a promising approach for MASs.

B. Counterarguments to more thorough evaluation

Even though evaluation is part of the standard scientific method there is no consensus in the amount of effort that should be put into it.

1) Thorough evaluation is too time consuming: A thorough evaluation is a very time consuming endeavor as we have painfully experienced in our attempts to do so. But most of the difficulty (and time) is in the fact that there are no standards for evaluation in dynamic logistics problems defined. Currently the efforts are scattered, there are few works that actually compare with multiple other approaches, which is understandable due to the considerable time it takes to do a good comparison. To reduce the time needed for evaluation we are strongly in favor of open sourcing all major components, such as simulators and benchmarks. We believe that only by making these components generally accessible and easy to use, researchers can truly benefit from previous efforts, or to
paraphrase Isaac Newton: with open source we can ‘stand on the shoulders of giants’.

2) A proof of concept is enough evaluation: The question one can ask himself is: why are multi-agent systems not a widely accepted paradigm in logistics even though there are countless papers that argue how well MAS fits the problem? We believe that the answer lies in the lack of thorough evaluation, a solid investigation of the performance of MAS in dynamic logistics problems is the only way to improve the scientific quality of MAS for logistics. Once the advantages and disadvantages of MAS in logistics are properly mapped this can only lead to a better understanding of MAS and ultimately to a more widespread adoption in industry.

IV. RELATED WORK

There are generally three branches of related work: the traditional centralized algorithms for dynamic logistics, multi-agent systems for logistics, and evolutionary designed multi-agent systems.

A. Centralized solutions for dynamic logistics

Pickup-and-delivery problems (PDP) [7], [8] are a wide area generally concerned with transportation of goods between locations. The dynamic variant [1], [9] is less researched, it is different from the static problem in that some of the input data is revealed or updated during the time operations take place. Two common strategies for handling dynamism are:

1) Continuous rescheduling: The naive approach to handling dynamism is to treat the problem as a sequence of static problems. For example, Attanasio et al. [10] implemented an algorithm that was used earlier for the static dial-a-ride problem. Their implementation was parallelized and tested on the dynamic dial-a-ride problem by sequentially executing it each time a new request arrived. Similarly in [11] a genetic algorithm was used, and Gendreau et al. [12] used neighborhood search heuristics for the dynamic PDP with time windows (commonly referred to as PDPTW). The impact of diversion (allowing vehicles to divert from their previous course) was investigated in [13], they show that allowing diversion can improve an existing solution.

2) Anticipation and flexibility: The following conjecture from [5] describes a slightly different approach: “if a problem requires sequential decision making under an uncertain future, and if the decisions impact the future state of the system, decision making should anticipate future needs”. An example can be found in [14] variable neighborhood search is used that aims for flexibility. The algorithm is flexible in the sense that it tries to make early decisions that provide flexible solutions that can later easily be adapted. Similarly, Bosman and La Poutré [15] use a probabilistic model of future requests in an evolutionary algorithm for dynamic vehicle routing. Ichoua et al. [16] adds ‘dummy’ demands to anticipate future demands in dynamic vehicle routing. Waiting strategies (postponing decisions) were researched in [17], [18].

B. Multi-agent systems for logistics

Multi-agent systems [19] are a widely researched distributed solution technique. It has been applied to logistics numerous times, in this section we highlight several noteworthy papers.

An influential paper that introduced a multi-agent system for dynamic transportation scheduling is [20]. It used the extended contract net protocol for cooperative transportation scheduling and it showed that its performance was comparable to existing operational research techniques, later the idea was combined with auctions in [21]. In [22] a MAS solution is investigated that also uses the contract net protocol, the solution was evaluated on real world data, the MAS produced better solutions than the solutions of professional dispatchers. A good comparison of a MAS and several traditional heuristics is presented in [23]. The MAS uses auction mechanisms for coordination and was shown to be at least similar in performance as the heuristics or better. The authors further claim that their MAS is more robust and flexible since it solves local problems locally. Måähr et al. [24] present a thorough evaluation of a MAS with auctions and a mixed-integer program on real world data of a PDPTW. Their results show that both solutions have comparable performance and that there is no significant difference. The results of the deployment of a MAS for a taxi company in London is reported in [25]. According to the paper the taxi fleet utilization has increased by 5 - 7 % since adopting the MAS.

C. Evolutionary designed multi-agent systems

Evolutionary computation is a well known computation paradigm inspired by natural evolution. It has been applied to a wide variety of engineering problems such as optimization, system control, machine learning and automated design [2]. There are many papers about evolving multi-agent systems outside of logistics. For example, Luke and Spector [26] present a MAS where the agents are programmed using genetic programming. The agents were lions in a virtual world that had to collaboratively hunt virtual ‘gazelles’. In other work, two agents were evolved to develop a communication protocol to find each other in an unknown grid world [27]. Evolution has also been applied to other domains such as teams of unmanned aerial vehicles [28]–[30], teams of (soccer) robots [31], [32] and groups of digital organisms [33], [34].

In the manufacturing domain, genetic programming has been used to develop dispatching rules in a job shop scheduling problem [35]. It was shown that the evolved heuristics outperformed the state-of-the-art hand constructed heuristics. In [36] an evolution strategy was used to evolve the weights of a dispatching rule for agents in a dynamic PDP. The dispatching rule was used by the agent and directly controls its behavior, the paper shows that it works but it is not compared to other solutions. Another application to dynamic PDP uses genetic programming [37]. Here, not only the weights but the entire structure of the rule was evolved. The authors compare their solution with a centralized selection hyper-heuristic. It is shown that the evolved MAS significantly outperforms the hyper-heuristic. In [38] genetic programming is used to as
a developmental solution to dynamic capacitated arc routing problems. A heuristic is evolved that indirectly controls a vehicle that has to travel over the road network. Their experiments on several benchmarks show that their approach is feasible on both the static and dynamic problem.

V. CURRENT STATUS AND FUTURE WORK

The two positions that we take in this paper are two research directions on which we have been working. In this section we present which challenges we have already (partially) overcome, and we discuss the next steps that we intend to take.

A. Current status

Since the research obstacles described in the previous section result from our experience in the domain we already started working on overcoming them. Recently we have introduced an open source [1] multi-agent simulator, called RinSim, specialized for logistics problems [39].

Further, we have studied the literature for dynamic pickup-and-delivery problems to investigate if any of the used datasets can be incorporated with our simulator as a benchmark. We are aiming to incorporate one or more of benchmarks used in [11],[12],[17],[40]. Besides we are collaborating with a European transport company and hope to be able to publish a benchmark using an anonymized version of their data.

Moreover, the well-attended Eval4SASO workshop (co-located with SASO-2013) and its lively discussions confirmed both the lack of disciplined evaluation approaches for self-organizing solutions today, as well as the need and strong interest for a common framework for scientific evaluation of such systems.

B. Future work

Here we present several interesting directions for future work for evolving multi-agent systems for logistics.

1) Improved validation: We believe that the work that was presented in [36]–[38] would benefit greatly from an improved evaluation. We plan to extend the work of [37] and validate it thoroughly on one or more benchmarks.

2) More complex behavior: Once the approach has been evaluated in detail we can start improving the evolved MAS. Several interesting avenues of research for producing more complex behavior are to incorporate grammars [41] or to use a neural network representation with NEAT [42].

3) Biological experiments in logistics: Experiments with different team compositions (homogenous or heterogenous) and different levels of selection (individual or team) as done in [43] could be conducted in the logistics domain. It would be interesting to see the effects of these choices under different circumstances in logistics problems.

4) Adaptive MAS: An interesting research topic would be to give offline evolutionary designed agents capabilities to learn and adapt online. There are many approaches to agent learning [44], we would like to see a learning approach that is possibly evolutionary designed. This can potentially be combined with the forecasting of dynamic events by agents.

VI. CONCLUSION

In this paper we argued that evolutionary designed multi-agent systems are a promising solution method for dynamic and scalable logistics problems. Further, we think that thorough evaluation of (multi-agent system) solutions should have a higher priority in the dynamic logistics field. We think that by pursuing our two positions progress in multi-agent systems for logistics will be accelerated and that it will be beneficial (since reusable) for the community at large.

ACKNOWLEDGMENTS

We would like to thank Rutger Claes for his effort in proofreading this paper. This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund K.U. Leuven.

REFERENCES


http://github.com/rinde/rinsim/


